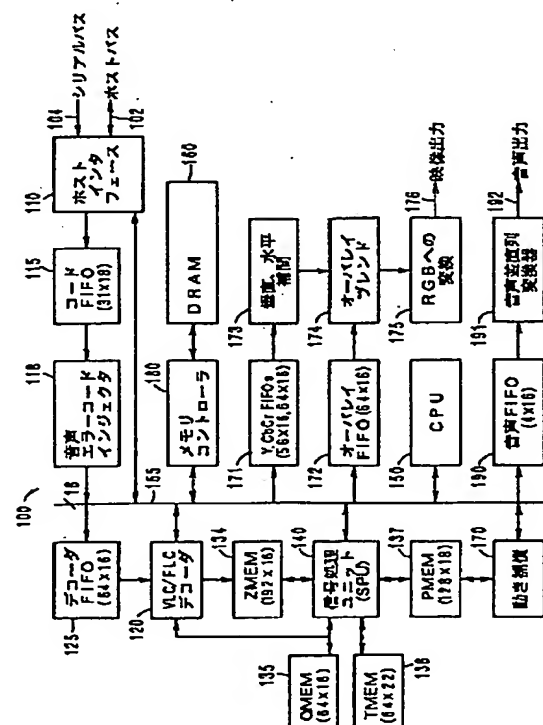


(11)特許出願公開番号



【特許請求の範囲】

【請求項 1】 第 1 内部メモリと、
音声データのブロックと映像データのブロックとを交互
に前記第 1 メモリに書き込む手段と、
前記第 1 メモリからの音声データのデコードと、前記第
1 メモリからの映像データのデコードとを交互に行う信
号処理ユニットとを有することを特徴とする一体化され
た音声映像デコーダ。

【請求項 2】 前記信号処理ユニットが、
映像データのデコードの間に逆離散コサイン変換 (I D
C T) 係数と量子化係数との乗算を行い、音声データの
デコードの間に量子化されたサンプルベクトルの要素と
逆量子化係数との乗算を行う乗算器と、
映像データのデコードの間に、逆離散コサイン変換のた
め I D C T 係数の差及び和を定め、音声データのデコー
ドの間に周波数を変数とするサンプルベクトルの要素の
差及び和を定めるバタフライユニットとを有することを
特徴とする請求項 1 に記載のデコーダ。

【請求項 3】 音声データフレームからサブバンドデ
ータをデコードして第 1 ベクトルを生成する過程であつ
て、前記第 1 ベクトルが、音声サンプルデータの周波数
を変数とする要素を表す要素を有する、該デコード処理
を行う過程と、
バタフライユニットを用いて、前記第 1 ベクトルの 2 つ
以上の要素を結合する過程と、
乗算器を用いて前記結合データとマトリクス処理係数と
の積を計算する過程と、
前記積をメモリロケーションに蓄積する過程と、
第 2 ベクトルの要素を定めるべく、前記結合過程、計算
過程、及び蓄積過程を一回以上繰り返す過程であつて、
前記第 2 ベクトルが、音声サンプルデータの周波数を変
数とする要素を表す要素を有する、該繰り返し過程とを
有することを特徴とする M P E G 規格音声データフレーム
をデコードする方法。

【請求項 4】 請求項 3 の結合過程、計算過程、蓄積
過程、及び繰り返し過程を複数回繰り返して、線形独立
な前記第 2 ベクトルの要素のセットを生成する過程と、
前記第 2 ベクトルの要素を第 2 メモリに格納する過程
と、
前記第 2 ベクトルの要素を第 2 メモリに格納する過程
と、
他のベクトルの要素を前記第 2 メモリから検索する過程
であつて、前記他のベクトルはそれぞれ音声サンプルデ
ータの周波数を変数とする要素を表す要素を有する、該
検索過程と、
前記第 2 ベクトルの要素と前記他のベクトルの要素とを
結合して音声振幅値を生成する過程とをさらに有するこ
とを特徴とする請求項 3 に記載の方法。

【請求項 5】 前記第 2 ベクトルの要素を格納する過
程が、互いに線形独立な要素のみを格納することを特徴

とする請求項 4 に記載の方法。

【請求項 6】 前記他のベクトルの要素を検索する過
程が、互いに線形独立な要素のみを検索することを特徴
とする請求項 5 に記載の方法。

【請求項 7】 時間を変数とするベクトルの要素を含
むブロックを第 2 メモリから第 1 メモリへ転送する過程
と、

前記ブロックの前記各要素の積を、対応するウィンドウ
処理係数によって定める過程と、

10 前記積を複数の合計値として蓄積する過程とを有し、
前記各合計値が異なった音声振幅値に対応することを特
徴とする M P E G エンコード規格に従ったデータから音
声振幅値を生成する方法。

【請求項 8】 前記ブロック転送過程が、第 1 の時間
を変数とするベクトルからの 1 7 個の要素と、第 2 の時
間を変数とするベクトルからの 1 6 個の要素とを転送す
る過程をさらに有することを特徴とする請求項 7 に記載
の方法。

【請求項 9】 前記積を定める過程が 6 4 個の乗算を
行い、各乗算は前記ブロックからの要素の 1 つとウィン
ドウ処理係数を含んだ計算であることを特徴とし、
前記蓄積過程が各 3 2 個の合計値の 1 対の積を加算する
過程を有することを特徴とする請求項 8 に記載の方法。

【請求項 1 0】 一連のマトリクス処理係数と、周波
数を変数とするベクトルの要素の対応する結合データと
の乗算を行う過程と、

時間を変数とするベクトルの 4 つの要素を生成するべく
積を蓄積する過程と、

30 前記 4 つの時間を変数とするベクトルの要素を前記第 2
メモリに書き込む過程とを有することを特徴とする請求
項 9 に記載の方法。

【請求項 1 1】 請求項 7、請求項 8、請求項 9、及
び請求項 1 0 の過程を 8 回繰り返し、かつ前記同じ 1 対
の時間を変数とするベクトルの要素をブロック転送する
2 つの過程は繰り返さないことを特徴とする請求項 1 0
に記載の方法。

【請求項 1 2】 第 1 マルチプレクサと、
第 2 マルチプレクサと、
動作時に、前記第 1 及び第 2 マルチプレクサから非除数
を表す信号を受け取るべく接続された入力バスを有する
除算器と、
余りを表す信号をうけとるべく前記除算器に接続された
入力バスと、前記第 1 マルチプレクサの第 1 入力バスに
接続された出力バスとを有する第 1 レジスタと、
商を表す信号を受け取るべく前記除算器に接続された入
力バスと、前記第 1 マルチプレクサの入力バスに接続さ
れた出力バスとを有する第 2 レジスタとを有することを
特徴とする逆グループ処理回路。

【請求項 1 3】 前記第 2 レジスタの前記出力バスに
接続された入力バスと、前記第 1 マルチプレクサの第 2

入力バスに接続された出力バスとを有するダイ 3 レジスタと、
前記第 1 レジスタの前記出力バスに接続された第 1 入力バスと、前記第 3 レジスタの前記出力バスに接続された第 2 入力バスとを有する第 3 マルチプレクサとをさらに有することを特徴とする請求項 1 2 に記載の逆グループ処理回路。

【請求項 1 4】 前記除算器が除数を選択するための選択ターミナルをさらに有することを特徴とする請求項 1 2 に記載の逆グループ処理回路。

【請求項 1 5】 前記除算器が、
前記第 1 及び第 2 レジスタの前記入力バスに接続されたデータバスを有するリードオンリメモリと、
前記除算器の前記入力バスと前記リードオンリメモリのアドレスバスとの間に接続されたアドレスジェネレータとをさらに有することを特徴とする請求項 1 4 に記載の逆グループ処理回路。

【請求項 1 6】 前記アドレスジェネレータが前記リードオンリメモリへアドレス信号を与え、前記選択ターミナル上の信号が第 1 の値を有するとき、前記アドレス信号が前記非除数を表す信号と等しく、前記選択ターミナル上の前記信号が第 2 の値を有するとき、前記アドレス信号が、前記非除数を表す信号と前記選択ターミナル上の信号との論理結合データと等しいことを特徴とする請求項 1 5 に記載の逆グループ処理回路。

【請求項 1 7】 前記第 1 マルチプレクサが前記アドレスジェネレータに第 1 の 4 ビット信号を与え、
前記第 2 マルチプレクサが前記アドレスジェネレータに第 2 の 4 ビット信号を与え、
前記アドレスジェネレータが前記リードオンリメモリに 8 ビットアドレス信号を与え、
前記選択ターミナル上の信号が第 1 の値を有するとき、
前記アドレス信号の 4 つの最上位ビットが前記第 1 の 4 ビット信号と等しく、前記アドレス信号の 4 つの最上位ビットが前記第 2 の 4 ビット信号に等しく、
前記選択ターミナル上の前記信号が第 2 の値を有する時、前記アドレス信号の最上位ビットが 1 に等しく、前記アドレス信号の最下位の 7 つのビットが前記第 1 及び第 2 の 4 ビット信号の最下位の 3 つのビットの結合データに等しく、
前記選択ターミナル上の信号が第 3 の値を有するとき、
前記アドレス信号の最上位ビットが 1 に等しく、前記アドレス信号の最下位の 7 つのビットが定数であるか、または前記第 1 及び第 2 の 4 ビット信号のそれぞれの前記 2 つの最下位のビットの結合データに等しいことを特徴とする請求項 1 5 に記載の逆グループ処理回路。

【請求項 1 8】 前記除算器が、
3 を除数とする除算回路と、
5 を除数とする除算回路と、

9 を除数とする除算回路と、
マルチプレクサとを有し、
かつ、前記マルチプレクサが、
前記選択ターミナルに接続された選択リードと、
前記除算器の前記入力バスに接続された入力バスと、
前記 3 を除数とする除算回路に接続された第 1 出力バスと、
前記 5 を除数とする除算回路に接続された第 2 出力バスと、

10 前記 9 を除数とする除算回路に接続された第 3 出力バスとを有することを特徴とする請求項 1 4 に記載の逆グループ処理回路。

【請求項 1 9】 データソースからのデジタルデータストリームをデコーダへ転送する過程と、
前記データソースがエラーを検出したとき前記データソースからのエラー信号を前記デコーダへ与える過程と、
前記エラー信号及びデータの一部が前記デコーダから受け取られたときデジタルデータストリームに於ける前記データの一部をエラーコードで置き換える過程と、
20 前記データストリームの中にあり前記エラーコードに等しいビット結合データを置き換えられるようにするべく前記デコーダにフラグ信号を与える過程と、
前記データストリームの中にあり、前記エラーコードと等しいビット結合データを置き換えることによって前記データストリームを変化させる過程と、
前記変化したデータストリームをデコードする過程とを有することを特徴とするエラーを含むデジタルデータストリームをデコードする方法。

【請求項 2 0】 前記エラーコードがエラーのないデータストリームに於ける有効なビット結合データであることを特徴とし、
前記フラグを立てたままにしておき、次いで、前記データストリームの中にあり前記エラーコードに等しいビット結合データを置き換えることを禁止するべく、前記フラグを立てないようにする過程をさらに有することを特徴とする請求項 1 9 に記載の方法。

【請求項 2 1】 前記エラーコードと等しいビット結合データがエラーのないデータストリームに於いて殆ど現れず、エラーのないデータストリームの中にありエラーコードと等しいビット結合データを置き換えることが、
40 デコードされたデータを殆ど変えることにならないことを特徴とする請求項 2 0 に記載の方法。

【請求項 2 2】 前記デジタルデータストリームが前記 MPEG エンコード規格に従った音声データストリームであることを特徴とする請求項 2 0 に記載の方法。

【請求項 2 3】 前記データストリームを変化させる過程がサブバンドデータを 0 に置き換える過程をさらに有することを特徴とする請求項 2 2 に記載の方法。

【請求項 2 4】 前記データストリームを変化させる
50 過程が、

前記エラーコードと等しいビット結合データがサブバンドデータ内にあるとき、前記サブバンドデータを0で置き換える過程と、

前記エラーコードに等しいビット結合データが前記ヘッダ内または前記第1データフレームのサイドインフォメーション内に少なくともその一部が存在するとき、第1音声データフレームを前音声データフレームで置き換える過程を更に有することを特徴とする請求項22に記載の方法。

【請求項25】 前記データストリームを変化させる過程が、前記ビット結合データを1またはそれ以上の前音声データフレームから導出される同形のデータで置き換える過程をさらに有することを特徴とする請求項22に記載の方法。

【請求項26】 前記エラー信号が与えられたとき、シフトレジスタ内の値に1を書き込む過程と、前記シフトレジスタ内の値を周期的にシフトさせる過程とを有し、前記フラグ信号を与える過程が、前記シフトレジスタ内の値が0でないとき前記フラグ信号を与えることを特徴とする請求項20に記載の方法。

【請求項27】 前記変化されたデータストリームをバッファに書き込む過程を有し、前記値をシフトする過程がエラーコードが前記バッファ内に残っている時間と同じかそれ以上の時間、0でないビットが前記シフトレジスタ内に残っているような速度で、前記値をシフトさせることを特徴とする請求項26に記載の方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本出願は、米国特許出願第07/669、818号の一部継続出願である米国特許出願第07/890、732号（平成5年特許願第151、595号）の継続出願である米国特許出願第08/288、652号の一部継続出願に基づくものであって、本発明は、デジタルデータから音声信号を生成するデコーダに関し、特に、MPEG規格に基づく結合された音声及び映像データのデコード処理に関する。

【0002】

【従来の技術】 モーションピクチャーエキスパートグループ（MPEG）は、デジタル媒体上の動画信号及びそれに関連する音声信号の圧縮及び復元に関する国際規格（以下、MPEG規格と呼ぶ）を開発した。国際標準化機構（ISO）による1993年に刊行の刊行物N o. 11172に於いて、「Coding for Moving Pictures and Associated Audio」という表題でMPEG規格について記載しているが、これを参照されたい。MPEG規格は、音声及び映像信号の符号化されたデジタル表現を特定し、コンパクトディスク、デジタルオーディオテープ、及び磁気ハードディスクのような装置からの1秒

当たり1.5メガビットの連続的なデータ伝送を予定している。

【0003】 MPEG規格では、平行データストリームまたは時間多重化データストリームが映像データフレーム及び音声データフレームを与える。映像データフレームのデコードシステム及びデコードの方法については、米国特許出願第07/890、7325（平成5年特許願第151、595号）及び米国特許出願第07/669、818号を参照されたい。音声データフレームはヘッダ、サイドインフォメーション（side information）、及びサブバンドデータを含む。サブバンドデータは周波数を変数とするベクトルを表し、このベクトルは変換処理（マトリクス処理）及び平滑フィルタ処理（ウィンドウ処理）によって時間を変数とする出力音声振幅値に変換される。

【0004】 典型的には、デコードのためのMPEG規格の音声映像デコードシステムは、一方は音声信号のデコード、他方は映像信号のデコードに用いる2つの集積回路チップ上に別々に設けられた2つのデコーダを有する。音声デコーダ及び映像デコーダはMPEG規格音符号化技術とMPEG規格映像符号化技術との差のために分離して設けられるが、音声映像信号デコーダは回路の量が増えるため、音声映像デコードシステムのコストを増加させる。デコードシステムのアーキテクチャは、回路の量を低減し、MPEG音声及び映像データのデコードに掛かるコストを減らすことを必要としているのである。

【0005】

【発明が解決しようとする課題】 従って、本発明の目的は、回路の量を減らした効率的な音声映像デコーダ及び音声映像デコードの方法を提供することである。

【0006】

【課題を解決するための手段】 本発明によれば、1つのチップ上に一体化されたMPEG音声映像デコーダが、音声及び映像信号の両方をデコードするべく、メモリ、主CPU、メモリコントローラ、及び信号処理ユニット（SPU）のような構成要素を用いる。SPUには乗算器（または乗算-蓄積ユニット（multiply-and-accumulate unit））及びバタフライユニットが含まれ、これらによって映像データ音声データのデコードを交互に実施する。乗算器とバタフライユニットとの結合によって、音声および映像信号のデコードを効率的に実施することができる。特に、音声信号のデコードのために、バタフライユニットで、周波数を変数とするベクトルの要素の和と差を決定することによって、マトリクス処理（即ち、周波数を変数とするサンプルベクトルから時間を変数とするベクトルの要素を決定すること）のために必要な乗算の数を減らすことが出来る。要素の結合データを定めることによって、結合された要素の逆量子化及び逆スケール処理も連続して実施することができる。更に、

マトリクス処理及びウィンドウ処理（即ち前時間を変数とするベクトルと現時間を変数とするベクトルとの結合データ）は1つの命令に統合されて、乗算器、バタフライユニット、及び外部メモリへの読み書きを行うメモリコントローラの平行利用を増やすことにより、デコーダのスループットを高めることができる。

【0007】また、本発明により、MPEG規格サブバンドコードをデコードするための逆グループ処理回路は除算器を有し、これは3つのクロック周期の間にMPEG規格サブバンドコードを3つのベクトル要素に転換する2つの除算を実行するものである。2つのクロック周期でなく3つのクロック周期で2つの除算を計算することにより、除算器は小さくコストの掛からないものとしてできるが、これによってデコードの速度が落ちるということはなく、これは3つのベクトル要素をシングルポートのメモリに書き込むのに3つのクロック周期が必要だからである。従って、除算器を小さくすることによってサブバンドコード逆グループ処理及び計算結果の要素をメモリに書き込むのに掛かる時間が著しく増加するということはない。除算器に入力される非除数に関する公知の限界を用いることによって、除算器の寸法及びコストは汎用除算器のそれと比較してずっと小さなものにすることができる。

【0008】更に本発明によれば、外部のMPEG音声データストリームソースからのエラー信号に応じて、MPEG音声デコーダはエラーコードを有する音声データストリームに置き換えるが、このエラーコードはMPEG音声データフレームに於いて殆ど見られないビット結合データであり、エラーの取り扱いを一時的に可能にすることが出来る。エラーコードを有する音声データストリームはデコーダに於いてセーブもしくはバッファリングされ得る。エラーの取り扱いを同時に可能にした音声のデコードの間に、デコーダはエラーコードと等しいビット結合データのための音声データを探し、再構築されたデータのビット結合データに置き換える。この置き換えによってエラーが音声化されるのを最小化することができる。典型的には、サブバンドデータの幾つかはゼロに置き換えられ、エラーによって周波数成分のいくらかが失われることになる。

【0009】

【発明の実施の形態】本発明によれば、音声映像デコーダはMPEG規格データストリームをデコードして、音声信号及び映像信号を与える。音声映像デコーダは映像データフレームのデコードと音声データフレームのデコードを交互に行い、両者に同じメモリ及び信号処理ユニット（SPU）が備えられている。

【0010】図1は、MPEG規格音声及び映像データフレームをデコードするための音声映像デコーダ100のブロック図を示したものである。デコーダ100はシリアルバス104を通してMPEG規格の符号化された

音声及び映像データを受け取り、そのデータをデコードして、デコードされたデータを映像バス176及び音声バス192を通して供給する。デコーダ100はスタティックランダムアクセスメモリ（SRAM）134～137（ここではZMEM134、QMEM135、TMEM136、及びPMEM137とも呼ばれる）を有し、これらは映像デコードのための映像データの保持及び音声デコードのための音声データの保持を交互に行う。デコーダ100はまた信号処理ユニット（SPU）140を有し、これは命令メモリ、レジスタファイル、乗算器または乗算-蓄積ユニット（MAC）、及びバタフライユニットを有しており、デコーダ100がその時点で映像のデコードを行っているか、或いは音声のデコードを行っているかに応じて、映像データ、または音声データのデコードを行う。

【0011】音声映像デコーダ100は、ホストバス102及びシリアルバス104を通して、ホストコンピュータ及びコンパクトディスクデジタル信号プロセッサ

（CD-DSP）のような音声及び映像信号ソースと接続されている。シリアルバス104はMPEG規格に従って圧縮された音声及び映像データのストリームを送るが、このデータストリームは先入れ先出し（FIFO）バッファ115（コードFIFO115）を通してデコーダ100が受け取るものである。メモリコントローラ180は、コードFIFO115からの圧縮されたデータをメインバス155を通して読み取り、その圧縮されたデータを外部メモリ160（ここではBRAM160とも呼ばれる）に書き込む。以下に述べるように、音声エラーコードインジェクタ118はエラーコードをDRAM160に書き込まれた音声データに注入することができる。CPU150はそれ自身の命令メモリを有するマイクロコードプロセッサであるが、これはメインバス155へのアクセスを制御し、特に、コードFIFO115からDRAM160へデータ伝送を行うメモリコントローラ180へコマンドを送る。

【0012】この実施例に於いて、DRAM160はダイナミックランダムアクセスメモリ（DRAM）コンポーネントを含む。また、他の適当なメモリ技術を用いることも出来る。DRAM160は、シリアルバス104からの圧縮されたデータ、及び音声バス192または映像バス176への出力のための復元されたデータを保持する。CPU150の制御のもとで、メモリコントローラ180は、圧縮された音声または映像データをデコーダFIFO125へ送り、SPU140によって音声データフレームまたは映像データのデコードを行う。

【0013】MPEG規格によれば、映像データフレームは、圧縮された画像のデジタル形式の記述であり、音声データフレームは最大2チャンネルの一定の数の、周波数を変数とする音声サンプルのデジタル形式の記述である。映像データフレーム及び映像データフレームをデ

コードして映像信号を生成するためのMPEG規格は、米国特許出願第07/890,732号(平成5年特許願第151,595号)及び第07/669,818号に記載されており、これらの明細書を参照されたい。MPEG規格は現在3つの音声データフレームの型を定めており、それらは層1、層2、及び層3データフレームと呼ばれる。図1のデコーダ100は層1及び層2の音声データフレームをデコードする。層1及び層2の音声データフレームはヘッダ、サイドインフォメーション

(side information)、及びサブバンドデータを含む。ヘッダが示すのは、音声データフレームを与えるデータストリームのビットレート、デコードされた音声のサンプル周波数、サブバンドデータのチャンネル数が1チャンネルであるか或いは2チャンネルであるか、及びサブバンドデータの音声チャンネルがそれぞれ独立か、ステレオか、またはインテンシティステレオ(intensity stereo)かを記述したモードエクステンション(mode extension)である。サイドインフォメーションが示すのはサブバンドデータのサブバンドごとに割り当てられるビット数、及び、以下に述べるように、サブバンドデータを逆量子化或いは逆スケール処理するためのスケールファクタFのインデックスである。

【0014】CPU150は、SPU140が音声データをデコードするのに費やす時間のパーセンテージを制御する。音声のデコードのために、CPU150はメモリコントローラ180を制御して、音声データをDRAM160からデコーダFIFO125に移動させ、またSPU140に音声データのデコードに必要な計算を実行させる。SPU140はCPU150と同時並行的に作動し、SPU140内の命令メモリに格納されたソフトウェアに従ってコマンドを実行する。

【0015】音声データフレームのデコード時に、SPU140は初めに「get bits」コマンドを実行するが、このコマンドは、デコーダFIFOバッファ125から、音声データフレームのヘッダ及びサイドインフォメーションを、VLC/FLCデコーダ120を通してCPU120へロードするものである。CPU150は、サイドインフォメーションからのビット数の割り当て及びスケールファクタをSPU140を通してQMEM135に書き込む。ヘッダ及びサイドインフォメーションは、変化せずにVLC/FLCデコーダ120を通過する。サブバンドデータは、デコーダFIFOバッファ125からのデータストリームに於けるサイドインフォメーションに従う。VLC/FLCデコーダ120は、映像データの可変長コード(VLC)をデコードし、音声及び映像データの固定長コード(FLC)をデコードするための回路を有する。VLC/FLCデコーダ120は以下に述べるように、音声データのための逆グループ処理回路も有する。

【0016】SPU140によって実行される「サブバ

ンド読み込み(get subbands)」コマンドは、VLC/FLCデコーダ120に、デコーダFIFOバッファ125からのサブバンドコードCiの構文解析を行い、スケール処理され量子化された要素Si192に転換させる。VLC/FLCデコーダ120は必要な逆グループ処理を行い、スケール処理され量子化された要素Si''をZMEM134に書き込む。各周波数を変数とするベクトルS''は32の周波数帯域(サブバンドi)に32の要素Si''を有する。「サブバンド読み込み」コマンドはZMEM134に要素Si''を書き込み、各チャンネルに3つの周波数を変数とするベクトルS''を与える(つまり2つのチャンネルに対して6つのベクトルS''が与えられる)。インテンシティステレオに対しては、周波数成分Siの幾つかは両チャンネルに共通に使用される。VLC/FLCデコーダ120は、各チャンネルに分配される要素のコピーを2つ書き込んで、ZMEM134に於ける各ベクトルS''が32の要素Si''を有することになる。モノラル音声に対しては、VLC/FLCデコーダ120は全ての要素Si''のコピーを2つ書き込むことが出来、それによってステレオの出力信号の両チャンネルが等しくなる。音声データフレームに於けるベクトルS''の数は、チャンネル数及び音声データフレームがMPEG規格の層1に従うか、層2に従うかによって決まる。層1の場合は、各チャンネルに12個のベクトルS''(384のサンプル)が存在する。層2に於いては、各チャンネルに36個のベクトルS''(1152のサンプル)が存在することになる。

【0017】SPU140は、「逆量子化/逆スケール処理(dequant/descale)」コマンドを実行して、ZMEM134からのSiを逆スケール処理及び逆量子化することにより周波数を変数とするベクトルSの要素Siを生成する。SPU140はTMEM136の一部にベクトルSの表現を書き込む。以下に記すようにマトリクス処理によって、周波数を変数とするベクトルSを、時間を変数とするベクトルVに変換する。SPU140は、時間を変数とするベクトルVの要素ViをPMEM137に格納し、メモリコントローラ180は、PME137からの要素ViをDRAM160に書き込む。PRAM160からの、一連の、時間を変数とするベクトルV⁰~V¹⁶の要素は、以下に述べるウィンドウ処理で結合され、この結合された要素はTMEM136に蓄積されることによって、32個の時間を変数とする出力音声振幅Aiを生成する。時間を変数とする出力音声振幅Aiは、典型的にはDRAM160の音声出力FIFOバッファに書き込まれ、DRAM160からメインバス155、出力音声FIFO190、及び音声並列変換器191を通して音声出力バス192に送られる。出力音声FIFOバッファ190は十分な出力音声振幅値を保持し、最も早いサンプリング速度に於いてもメインバス155へのアクセスの遅れによって音声干渉され

ることがない。音声並直列変換器191は出力音声データを連続したデータストリームに変換し、D/A変換器(DAC)及びアンプ(図示せず)が音声データから音声を出力する。

【0018】サイドインフォメーションは、サブバンドiに於ける各量子化された要素S_i (及び各サブバンドコードC_i)のとりことが可能な数値の数を示す。例えば、サブバンドiに於けるサブバンドコードC_iが0, 2, 4, . . . , 2^Nの値を有する場合、各コードC_iには0, 1, 2, . . . , Nビットが用いられる。サブバンドiに用いられているビット数が0の場合は、VLC/FLCデコーダ120が要素S_iに対してZMEM134に0を書き込み、ベクトルSの有する要素の内0でない要素が32個に及ばないことになる。サブバンドiに対する2^N個の可能な数値を表現するビット数の分配に対しては、VLC/FLCデコーダ120はQM*

$$C_i = 3^2 \cdot S_{3i''} + 3 \cdot S_{2i''} + S_{1i''} \quad (\text{式1})$$

【0021】同様に、式2によって与えられる7ビットのコードC_iは、それぞれ0~4の5つの値をとり得る3つの要素S_{1i''}、S_{2i''}、S_{3i''}を表す。 ※20

$$C_i = 5^2 \cdot S_{3i''} + 5 \cdot S_{2i''} + S_{1i''} \quad (\text{式2})$$

【0023】式3は、それぞれ0~8の9つの値をとり得る3つの要素S_{1i''}、S_{2i''}、S_{3i''}を表す10ビットのコードC_iを与える。 ☆

$$C_i = 9^2 \cdot S_{3i''} + 9 \cdot S_{2i''} + S_{1i''} \quad (\text{式3})$$

【0025】VLC/FLCデコーダ120は、コードC_iを、式1~式3で与えられる3つの要素S_{3i''}、S_{2i''}、S_{1i''}に逆グループ処理し、スケール処理され量子化された要素S_{3i''}、S_{2i''}、S_{1i''}をZMEM134に書き込む。式1~式3で与えられるコードC_iの逆グループ処理には、2つの除算を行えば ☆

$$(C_i/x) = Q_1 = x \cdot S_{3i''} + S_{2i''}$$

【0027】もう1度xで除することによって、商Q₂及び余りR₂が式5によって与えられる。 ◇

$$(Q_1/x) = Q_2 = S_{3i''}$$

【0029】ZMEM134がVLC/FLC120に接続されたポートをただ1つ有する場合、要素S_{3i''}、S_{2i''}、S_{1i''}を書き込むのに必要なクロック周期は3クロック周期である。従って、VLC/FLCデコーダ120が、C_iを逆グループ処理するための2つの除算を行うのに掛かる時間は3クロック周期となり、要素S_{3i''}、S_{2i''}、S_{1i''}を書き込むパイプライン処理の遅れは起こさずにすまずことができる。

【0030】図2に示すのは、3クロック周期の間にコードC_iを逆グループ処理するために2つの除算を行うデコード回路200である。第1の除算は、2クロック周期の時間がかかる除算である。第2の除算は、1クロック周期の間に行われる。始めの除算に2クロック周期

*EM135に於けるサイドインフォメーションからのビットアロケーションを用いて、データストリームに於ける要素S_iの開始から終了までを同定し、要素S_iをZMEM134のワード数が整合された位置に書き込む。

【0019】MPEG規格によって、要素S_{i''}が3か5か9の値をとり、サブバンドiからの3つの異なるベクトルS₁、S₂、及びS₃の3つの要素S_{1i''}、S_{2i''}、S_{3i''}を符号化し、1つのコードC_iにすることができることになる。例えば、量子化されスケール処理された3つの要素S_{1i''}、S_{2i''}、及びS_{3i''}は、それぞれの可能な値が0か1か2である場合は、27種類の結合データが生成可能である。式1によって与えられる5ビットのサブバンドコードC_iは27個の可能な結合データを表す。

【0020】

【数1】

※【0022】

【数2】

☆【0024】

【数3】

☆分である。例えば、式C_i = x² · S_{3i''} + x · S_{2i''} + S_{1i''}と要素S_{3i''}、S_{2i''}、S_{1i''}がxより小さい場合は、C_iをxで除することによって、式4により商Q₁及び余りR₁が与えられる。

【0026】

【数4】

$$\text{余り } R_1 = S_{1i''} \quad (\text{式4})$$

◇【0028】

【数5】

$$\text{余り } R_2 = S_{2i''} \quad (\text{式5})$$

の時間をかけるようにすることによって、より小型の除算器を使用することができるようになり、VLC/FLCデコーダ120のコストを下げるができる。図2の実施例に於いては、除算器210は被除数をマルチプレクサ220及び221から受け取り、それを3、5、または9の値を持つ除数Xで除して商Q及び余りR_{out}を得る。サイドインフォメーションは各サブバンドにビット数を割り当て、逆グループ処理を必要とする各サブバンドに対して除数Xの値を定める。

【0031】コードC_iは逆グループ処理の始めの除算のために3つの部分C_{iH}、C_{iM}、及びC_{iL}に分割される。C_{iL}は、除数Xが3、5、または9のそれぞれの値を取る場合に対して2、3、または4つの最下位

のコード C_i のビットを含む。同様に C_{iM} は除数 X が 3、5、9 の値をとるそれぞれの場合に対して、より上位のコード C_i のビットを含み、 C_{iH} は 1、2 ビットのコードを C_i を含む。 C_{iH} は、0、2、3 または 4 ビット左側にパディングされる。

【0032】逆グループ処理は以下のように行われる。第 1 のクロック周期に於いて、マルチプレクサ 220 及び 221 は、信号 C_{iH} 及び C_{iM} を除算器 210 に与え、除算器 210 は商 Q_{1H} 及び余り R_1' を生成し、これらの商と余りは第 1 のクロック周期の終わりにレジスタ 231 及び 230 に書き込まれる。図 2 の実施例に於けるレジスタ 230 及び 231 は、エッジトリガされたデバイスであるが、別の実施例として、レジスタ 230、231 はラッチ、メモリロケーション等、与えられたデジタルデータ信号を保持できるデバイスであるものも可能である。第 2 のクロック周期の間に、マルチプレクサ 220 及び 221 は、それぞれレジスタ 230 からの余り R_1' 及び信号 C_{iL} をそれぞれ除算器 210 に与え、除算器 210 は商 Q_{1L} 及び余り R_1 を発生する。第 2 のクロック周期の前に、商 Q_{1L} 及び余り R_1 はそれぞれレジスタ 231 及び 230 に書き込まれ、商 Q_{1H} はレジスタ 231 からレジスタ 232 に書き込まれる。商 Q_{1H} 及び Q_{1L} はそれぞれ式 4 に於いて最上位と最下位の商 Q_1 のビットである。余り R_1 は式 1、2、3 のように値 S_{1i} とされる。

【0033】第 3 クロック周期の間に、マルチプレクサ 220 及び 221 は、それぞれレジスタ 230 及び 231 から信号 Q_{1H} 及び Q_{1L} を除算器 210 に与える。除算器 210 は商 Q_2 及び余り R_2 を発生し、それらの商と余りは式 5 で与えられ、またマルチプレクサ 240 はレジスタ 230 から値 R_1 を選択して、図 1 の AMEM134 のようなメモリに書き込む。第 3 のクロック周期の終わりに、商 Q_2 及び余り R_2 はレジスタ 231 及び 230 に書き込まれ、商 Q_{1L} はレジスタ 231 からレジスタ 232 に書き込まれる。

【0034】第 4 のクロック周期に於いて、 S_{2i} と等しい余り R_2 は、マルチプレクサ 240 を通過してメモリに書き込まれる。商 Q_2 は、第 4 のクロック周期の終わりにレジスタ 232 に書き込まれる。 S_{3i} と等しい商 Q_2 は、第 5 のクロック周期の間にメモリに書き込まれる。第 2 のコード C_i' のための第 1 の除算は、第 4 及び第 5 のクロック周期の間に行うことができ、上記のように処理することができる。従って、一連のコード C が逆グループ処理される場合、逆グループ処理に伴うパイプライン処理の遅れは、一連のコードの始めの部分だけに限られることになる。

【0035】除算器 210 には従来のデジタル除算器回路が備えられ、除算器は除算器回路に正しい値の被除数、商、及び余りを取り扱うようにさせる。図 3 に示すのは、3 を除数とする除算回路 300 のブロック図であ

り、この回路はコード C の値の大きさの限界を利用して必要なゲート及びトランジスタの数を減らすものである。3 を除数とする除算回路 300 は、2 つの同等な回路ブロック 310 及び 320 を有する。各ブロック 310 及び 320 は 1 ビットの被除数信号 C 及び 2 ビットの余り信号 R_{in} のための入力ターミナル、及び 1 ビットの商信号 Q 及び 2 ビットの余り信号 R_{out} のための出力ターミナルを有する。ブロック 310 からの出力余り信号 R_{out} はブロック 320 の入力余り信号 R_{in} となる。図 2 の除算器 210 が用いられる場合、3 を除数とする除算回路 300 は、マルチプレクサ 220 (図 2) に接続された余りバス 312 と、マルチプレクサ 221 に接続された被除数バス 311 及び 321 と、レジスタ 230 に接続された余り出力バス 324 と、レジスタ 231 に接続された商出力バスとを有する。マルチプレクサ (図示せず) は、サブバンドのビット割当に応じて 3 を除数とする除算回路 300 を選択するような接続を与える。

【0036】図 4 に示すのは、回路ブロック 310 及び 320 のそれぞれに対する入力信号 R_{in} 及び C と、出力信号 R_{out} 及び Q との真理値表である。第 1 クロック周期の間、バス 312 に与えられる最大入力信号 R_{in} と 01 である最大信号 C_{iH} とは等しく、これはコード C_i は 5 つのビット数しか有していないからである。3 を除数とする除算に対する余りの最大値は 2 進法で 10 であり、式 4 の商 Q_1 の最大値は 2 進法で 0100 である (26 を 3 で割ると 8 余り 2 である)。従って、 C_{iH} に等しい入力信号 R_{in} 最もビット数の大きい Q_1 、または余りは 2 進法で 11 にはならないはずである。

【0037】図 4 の真理値表は、出力信号及び Q 及び R_{out} を示すことによって、信号 R_{in} が 11 を表す可能性を減らしていることを示している。残りのエントリは 3 ビットの値を 3 で除すことによって生成される。図 5 は、図 4 の真理値を実現したゲートレベルの図を示したものである。入力信号 A 、 B 、 C を反転するインバータをカウントすると、図 5 の回路を実現するためには、ブロック 310 または 320 に対して 30 個のトランジスタが必要である。図 4 の真理値表は他の様々な回路によっても実現することができる。

【0038】図 6 に示すのは 5 を除数とする除算回路 400 のブロック図であって、この回路はコード C の値が限られていることを利用して 5 で除すのに必要なゲート及びトランジスタ数を減らしている。5 を除数とする除算回路 400 は 3 つの同等な回路ブロック 410、420、及び 430 を有し、各ブロックは 3 ビットの余り信号 R_{in} 及び 1 ビットの被除数信号 P_{in} のための入力ターミナルと、3 ビットの出力余り信号 R_{out} 、及び 1 ビットの商信号 Q のための出力ターミナルとを有する。ブロック 410 の出力余り信号 R_{out} はブロック

10

20

30

40

50

420の入力余り信号Rinとなり、ブロック420の出力余り信号Routは、ブロック430の入力余り信号Rinとなる。

【0039】図2の除算器210に於いて用いられる場合、5を除数とする除算回路400は、マルチプレクサ220（図2）に接続された余りバス412と、マルチプレクサ221に接続された被除数バス411、421、431と、レジスタ230に接続された余り出力バス434と、レジスタ231に接続された商出力バス413、423、433とを有する。マルチプレクサは各サブバンド要素毎の5つの値のビット数の割当てに応じて、5を除数とする除算回路400を選択する接続を与える。

【0040】図7に示すのは、各ブロック410、420、及び430に対する可能な入力信号Rin及び例と所望の出力信号Q及びRoutとの関係についての真理値表である。図8は、図7の真理値表を実現する回路のゲートレベルの図である。図に示す真理値表を実現する回路に於いては、各回路ブロック410、420、及び430に対して48個のトランジスタが必要である。本発明によれば、図7の真理値表を実現する他の回路を用いることも可能である。

【0041】図9に示すのはデコード回路500であって、これはROM550とアドレスジェネレータ560を用いて、9、5、または3を除数とする除算を行う。2つの制御信号Div3及びDiv5がハイレベルであることは、それぞれ3を除数とする除算、及び5を除数とする除算を表している。2つの信号Div3及びDiv5がローレベルであるときは、信号R[3:0]及びD[3:0]によって表される8ビットの被除数が9で除算される。9を除数とする除算を行うために、信号R[3:0]、D[3:0]はアドレスジェネレータを560を変化せずに通過し、アドレス信号A[7:0]をROM550に与える。ROM550は、4つの最上位のビットと、8ビットの被除数R[3:0]、D[3:0]を9で除すことによって生ずる余り及び商に等しい最下位の4つのビットとを保持する。ROM550は2進法で0から10001111の間の被除数のために、アドレス00000000から10001111に145個の8ビットの値を保持する。このような範囲を取ることによって、余りR[3:0]の最大値が1000

(8)であるため、9で割られる被除数を全て用い尽くすことになる。

【0042】Div5がハイレベルであるとき、除算器210は6ビットの商を5で除算する。信号R[3:0]及びD[3:0]に於ける入力ビットR3及びD3は常に0であり、入力ビットR0、R1、R2、D0、D1、D2は6ビットの商を与える。図10に示すように、アドレス信号A[7:0]から1、R2、R1、R2、R0、D2、D1、D0のアドレスビットA7か

らA0をそれぞれセットする論理回路を有する。ここでRR2は要素R2の補数である。これによって信号R[2:0]及びD[2:0]によって与えられる、全ての6ビットの被除数に対するアドレスが与えられる。5を除数とする除算のためのアドレスは、2進法で10001111より大きいので、このアドレスは9を除数とする除算のためのアドレスにオーバーラップしない。

【0043】Div3がハイレベルであるとき、除算器210は4ビットの商を3で除算する。入力ビットR3、R2、D3、D2は、0であり、入力ビットR0、R1、D0、D1は4ビットの商を与える。図10に示すように、アドレスジェネレータ560は、信号Div3がハイレベルであるとき、アドレス信号A[7:0]のビットA7~A0を、1、0、1、0、R0、R1、D1、D0へセットする論理回路を有する。全ての4ビットの商に対するアドレスが、これによって与えられる。3を除数とする除算のためのアドレスは2進法で10001111より大きく、0であるアドレスビットA6及びA4を有する。従って、3を除数とする除算のアドレスは、10001111より小さい、9を除数とする除算のアドレス、またはアドレスビットA4の相補ビットであるアドレスビットA6を有する、5を除数とする除算のアドレスとオーバーラップしない。

【0044】ROM550は、199個の8ビット出力値を有する。必要なアドレスの最大値は2進法で11000111であり、ここには100111を5で除した余り及び商が含まれる。余り信号R[2:0]の最大値は、5を除数とする除算のためには2進法で100

(4)であり、商の最大値を制限する。

【0045】図1のVLC/FLCデコーダ120は、デコーダFIFOバッファ125からのデータストリームのサブバンドコードCiの同定、及び逆グループ処理を行い、量子化及びスケール処理された値Si''をZMEM134に書き込む。ZMEM134は、6つのサンプルベクトルS''の要素を保持するのに十分な192の16ビットワードを格納できる。サブバンドコードCiが、3つのベクトルS1''、S2''、及びS3''に対して値S1i''、S2i''、及びS3i''を与える場合は、2つのチャンネルのためのサブバンドコードは、MPEG規格の下ではデータストリームに於いてインタリーブされるので、記憶された192ワードが必要となるのである。従って、最悪の場合に於いては、3つの異なるベクトル（各チャンネル毎に3つ）の要素は、1つのベクトルS''が全てわかる前にデコードされなければならないことになる。

【0046】図1の実施例に於いては、SPU140がサブバンドデータを読みとる命令をVLC/FLCデコーダ120に出した後、VLC/FLCデコーダ120はサブバンド毎のビットアロケーションをQMEM135から読み取り、サブバンドコードCをデコーダFIF

0バッファ125から読み取り、(必要ならば)サブバンドコードCを逆グループ処理し、6つのベクトルのスケール処理され量子化された要素 S_i'' をZMEM134に書き込むが、これら全てはSPU140からの介入なしに行われる。このため、SPU140は他のオペレーションを行うために解放されることになる。他の実施例に於いては、SPU140が、サブバンドコードの読み取り及び逆グループ処理の制御により専念できる。例えば、SPU140は各サブバンドのビットアロケーションを読み取ることができ、VLC/FLCデコーダ120は、SPU140からの分離コマンドに応じて各サブバンドを逆グループ処理することができる。SPU140がデコーダFIFOバッファ125からの値を逆グループ処理する場合、VLC/FLCデコーダ120は*

$$S_i' = K1 \cdot (S_i'' + K2)$$

(式6)

【0049】

※ ※【数7】

$$S_i = F \cdot S_i'$$

(式7)

【0050】ここで、定数K1及びK2は値 S_i'' を表すのに用いられるビット数によって決まり、音声データフレームのサイドインフォメーションはスケールファクタFを表すインデックスを与える。式6に値 S_i'' を与えることを逆量子化と呼ぶこともある。式7に値 S_i' を与えることを逆スケール処理と呼ぶこともある。逆量子化及び逆スケール処理されたベクトルSの要素 S_i の結合データは以下に述べるように結合されてTMEM136に格納される。TMEM136は、64の22ビット記憶ロケーション、ベクトルSのための32の記憶ロケーション、及び以下に述べるように32の出力音声振幅 A_i のセットを蓄積するための32のロケーションを☆

$$N_{ji} = \cos[(16+j)(2i+1)\pi/64] \quad (式8)$$

【0053】ここで $0 \leq j \leq 63$ 、 $0 \leq i \leq 63$ である。マトリクスNは参考資料Aに示す31の異なる正の値、A~Z及びAA~AEのみを含む 32×64 のマトリクスである。マトリクスNの残りの要素は0または-1であるが、値A~ZまたはAA~AEの1つの符号を負にしたものである。式8は値A~Z及びAA~AEを☆

$$V_j = \sum_{i=0}^{31} N_{ji} \cdot S_i$$

(式9)

【0056】ベクトルVの32の要素 V_j は線形独立である。

【0057】V16及びV48を除き、ベクトルVの各要素 V_j を求めるには、式9に従って、マトリクスNの要素 N_{ji} とベクトルSの要素 S_i との32個の積を計算することが必要である。(V16は常に0であり、V48は要素 S_i の合計値の符号を負にしたものである。)要素 $S_0 \sim S_{31}$ の和と差である $T_0 \sim T_{31}$ を求めるためにバタフライユニットを用いることにより、乗算の数が減ることになる。ベクトルVを生成するのに必要な乗算の数を減らす結合データ $T_0 \sim T_{31}$ の例は参考資料Bに示されている。参考資料Cに示す

*取り除くこともできる。VLC/FLC120がサブバンド読み込み(「get subbands」)コマンドを実行しているとき、ZMEM134には6つのベクトルより大きな空間が空いていないため、SPU140はアイドル状態にある。しかし、上記のように、VLC/FLCデコーダ120は、SPU140を用いる場合よりも高速で逆グループ処理を行い得るものであり、従ってデコーダ100のスループットを高めるものである。

【0047】スケール処理された、及び量子化された要素 S_i'' は、式6及び7に示すように要素 S_i に対応する。

【0048】

【数6】

☆含む。

【0051】TMEMのサンプルベクトルSからの出力音声振幅 A_i を定めることにより、ここでマトリクス処理及びウィンドウ処理と呼ばれる処理ステップが実行されることになる。マトリクス処理は、MPEG規格によって与えられた変換マトリクスNと周波数を変数とするサンプルベクトルSとの積を計算することによってベクトルVを得ることを指す。マトリクスNの要素 N_{ji} は以下のように求められる。

【0052】

【数8】

☆示す。

【0054】ベクトルVは式9で与えられる64の要素 V_j を有する。

【0055】

【数9】

のは結合データ $T_0 \sim T_{31}$ によってベクトルVの要素 $V_0 \sim V_{63}$ を与える式である。 $T_0 \sim T_{31}$ を計算するのに用いられる要素 S_i 及び中間値は、SPU140のレジスタファイルに一時的に格納される。

【0058】参考資料Cの式は式9に等しいが、これは乗算の数が少なく済み、従ってデジタルシステムに於いてより高速の計算を行うことができる。ベクトルVに於いて、要素V16は常に0である。4の倍数のインデックスを持つ要素 $V(4n)$ は、それぞれ4つの乗算が必要である。4の倍数+1または3のインデックスを持つ要素 $V(4n+1)$ または $V(4n+3)$ は、それぞれ16の乗算を必要とする。4の倍数+2のインデッ

クスを持つ要素 $V(4n+2)$ は、それぞれ8つの乗算を必要とする。要素 $V0 \sim V15$ 及び $V49 \sim V63$ はそれぞれ等しい値か、要素 $V32 \sim V17$ 及び $V47 \sim V33$ の符号を負にしたものである。図7に示すSPU140、MAC750に於いて、結合データ $T0 \sim T31$ とROM732に格納された量との乗算によって、ベクトル V の要素 $V17 \sim V48$ のような32の線形独立の要素が求められる。要素 $V17 \sim V48$ はPMEM137に格納され、 $V48$ から $V17$ に至る順番で外部DRAM160に書き込まれる。別の実施例に於いては、他の線形独立な要素の組や異なった順番を用いてもよい。以下に述べるように、DRAM160に要素が書き込まれる順番は、ウィンドウ処理がいかに実行されるかを決定する。

【0059】1つのチャンネルのための連続したベクトル*

$$A_i = \sum_{k=0}^{15} U(i+32k) \cdot D(i+32k) \quad (\text{式10})$$

【0062】ここで $0 \leq i \leq 31$ である。またベクトル U は、 $U(64m+i) = V^{-2m} i$ によって定められる第1の要素のセット、及び $U(64m+32+i) = V^{-(2m+1)}(32+i)$ によって定められる第2の要素のセットを有し(ここで $0 \leq m \leq 7$)、また $D(i+32k)$ は、MPEG規格によって与えられる512個の要素ベクトルの要素である。式10では、ウィンドウ処理のために、16個のベクトル $V^0 \sim V^{-15}$ の64個の要素を必要とするが、これはたとえ各ベクトルが32個の独立した要素しか持っていない場合でも同じである。DRAM160に独立した要素のみが格納されていることによって、DRAMの必要なサイズが小さくなり、マトリクス処理の後の要素の書き込みにかかる時間が短くて済み、またウィンドウ処理のために要素を検索する時間も短くて済む。

【0063】参考資料Cに示すように、各ベクトル V は32個の線形独立な要素を有する。要素 $V1 \sim V15$ はそれぞれ要素 $V17 \sim V32$ の符号を逆にしたものに等しい。要素 $V16$ は0である。要素 $V33 \sim V47$ はそれぞれ要素 $V63 \sim V49$ に等しく、 $V48$ は独立である。要素 $V48 \sim V17$ のような線形独立な集合のみが、式10の2つの値 $D(i+32k)$ と要素との積を計算する場合ウィンドウ処理に必要となる。

【0064】式10に見られるように、各ベクトル V^{-2m} からの第1の32個の要素 $V^{-2m} i$ の1つ及び各ベクトル $V^{-(2m+1)}$ からの最後の32個の要素 $V^{-(2m+1)}(32+i)$ の1つは出力音声振幅 A_i に寄与する。各ベクトル V の第1の32個の要素は、独立値を16個だけ含む。各ベクトル V の最後の32個の要素は、17個の独立値を含む。従って、2つのベクトル V^{-2m} 及び $V^{-(2m+1)}$ からの合計33個の値は、ウィンドウ処理の間に、32個の出力音声振幅 A_i へのベクトル V^{-2m} 及び

*ル S は、音声データフレームの1つのためのデータストリームから順番にベクトル V に転換される。ベクトル V (または S)の順番を区別するために、ここでは肩文字が用いられることもある。ベクトル S^0 から求められたベクトル V^0 の要素は、DRAM160に於いてそのチャンネルの最も古いベクトルの要素の上に書き込まれる。ベクトル V^0 が求められている間、15個のベクトル $V^{-1} \sim V^{-15}$ の要素はDRAM160にある。ウィンドウ処理によって、今求められたベクトル V^0 からこれに続く15個のベクトル $V^{-1} \sim V^{-15}$ の要素を結合することにより、出力音声振幅値 A_i が求められる。

【0060】MPEG規格に基づき、以下の式により32個の出力音声振幅 A_i が与えられる。

【0061】

【数10】

$V^{-(2m+1)}$ の寄与を定めるのに十分なものである。この33個の値をDRAM160の連続的なアドレスに格納することによって、ウィンドウ処理のために値を読み取る速度が高められるが、これは、連続的なアドレスにアクセスする場合はページ変更の回数が最小で済むからである。

【0065】図11に示すのは、図1のDRAM160の一部のための、3つの別のメモリマップ610、620及び630である。メモリマップ610に於いては、ベクトル $V^0, V^{-1}, V^{-2}, \dots, V^{-15}$ がインデックスの小さい順に連続的に要素と共に格納される。式10に基づくウィンドウ処理のために、32の出力振幅 A_i を決定するには、各ベクトル $V^0, V^{-2}, \dots, V^{-14}$ の始めの32個の要素、及び各ベクトル $V^{-1}, V^{-3}, \dots, V^{-15}$ の最後の32個の要素が必要である。ウィンドウ処理のために要素を読み取るには、64の連続的なアドレスの7つのブロック612、613、 \dots 、918、及び32個の連続的なアドレスの2つのブロック611、619からの転送が必要となる。典型的には、ブロック611 \sim 619からの7つの転送では、それぞれ少なくとも1回は新しいページへのアクセスが必要となる。典型的なDRAMに対しては、ページの変更が必要な転送はアドレッシングのために別のクロック周期が必要であり、従って現在ページからの転送よりも長い時間がかかることになる。メモリマップ620に示すように、インデックスを小さくしていく順番にベクトル $V^0 \sim V^{-15}$ の要素を格納することによって、64の連続的なアドレスの8つのブロック621 \sim 628への転送の回数を減らすことになる。

【0066】メモリマップ630は32個の線形独立な要素 V_i のみを格納するが、ここでインデックス i には、ベクトル $V^0 \sim V^{-15}$ に対してそれぞれ48 \sim 17が

与えられる。別の実施例に於いて、線形独立な要素のセット及び要素の順番は、ウィンドウ処理係数 $D(i+32k)$ に適当に整合する場合は変更されても良い。メモリマップ 630 に対しては、ウィンドウ処理には 33 の連続したアドレスの 8 つのブロック 631~638 からの転送が必要であり、転送されるバイト数はほぼ半分にカットされる。この処理の実行に於いては、33 のベクトル要素の少なくとも 1 つのブロックは連続的なアドレスとならないが、これは現ベクトル V^0 がメモリに於いて 16 個の位置を占めており、メモリマップ 630 に示されているように最小のアドレスを必ずしも占めるものでないからである。何れにせよ、メモリマップ 630 からの転送はメモリマップ 620 からの転送よりもページ変更の機会を含む傾向が少なくなるが、これはブロック転送 631~638 がブロック転送 621~628 よりも小さいからである。

【0067】図 1 に戻ると、ベクトル $V^0 \sim V^{15}$ の要素は DRAM 160 のメモリマップ 630 (図 11) から PMEM 137 に転送される。転送された要素は式 10 のウィンドウ処理係数 $D(i+32k)$ と掛け合わせられ、これによって得られた数値が TMEM 136 に蓄積される。16 個のベクトルからの寄与が一旦蓄積されると、ウィンドウ処理は完了し、TMEM 136 に蓄積された 32 個の音声振幅値 A_i は DRAM 160 内の音声出力 FIFO バッファに書き込まれる。

【0068】SPU 140 は、ZMEM 134 の全てのサンプルベクトル S に対して逆量子化、逆スケール処理、マトリクス処理、及びウィンドウ処理を行う。そして次に「サブバンド取り込みコマンド」に応じて ZMEM 134 に更にベクトルがロードされる。音声データフレームの全てのサブバンド情報を読み込んだ後、SPU 140 が MPEG 映像データフレームのデコードに用いられる一方で、D/A 変換器 (図示せず) は DRAM 160 から音声出力 FIFO バッファ 190 を通してデコードされた音声振幅を読み込む。

【0069】デコード 100 の回路ブロック 170~175 は映像のデコードを実現する。ブロック 170 は、MPEG 規格に基づき、エラータームに基準ブロックを追加すると共に、半画素オフセット処理を行う。ブロック 171~175 は音声映像デコード 100 の映像出力ユニットを形成する。メモリコントローラは DRAM 160 から映像 FIFO 171 にデコードされた映像データを転送する。補間器 173 はデコードされた映像データを垂直及び水平にそれぞれ 2 倍に拡張する。リリックテキスト (lyric text) を表すデータのような映像オーバーレイデータは、DRAM 160 から読み込まれ、オーバーレイ FIFO 172 に与えられ、ブロック 174 は、補間器 173 からのデータとオーバーレイデータとをブレンド処理して画素値を供給する。コンバータ 175 は所望に応じて画素値を YCbCr カラー表現から RGB カラー表現に転換するが、これは映像出力バス 176 を通して伝送される。

【0070】図 12、図 13、図 14、及び図 15 は本発明に基づく信号処理ユニット (SPU) 140 のブロック図を示す。SPU 140 は命令メモリ (図示せず)、コントロールユニット (図示せず) を有し、コントロールユニットは命令メモリに格納されたデコードプログラムを実行する。SPU 140 は、メモリ 134~137 に格納された情報を用いて音声及び映像データフレームをデコードする。図 12、図 13 は音声デコードのための SPU 140 の部分を示す。ZMEM 134 は $(3 \times 64) \times 16$ ビット SRAM であり、音声デコードの間に、32 の 16 ビット要素 S_i をそれぞれ含む 6 つのベクトル S を格納するだけの大きさを有する。映像デコードの間に、ZMEM 134 は 2 または 3 セットの 64 の 9 ビット映像係数を格納するジグザグメモリとして機能する。QMEM 135 は $64 \times (2 \times 8)$ のメモリである。音声デコードの間、QMEM 135 は 2 つの音声チャンネルに対して 32 のサブバンドビットアロケーション及びスケールファクタのインデックスを保持する。映像デコードの間、QMEM 135 は、MPEG 規格に従って映像量子化マトリクスの 64 個の 8 ビット要素のセットを保持する。量子化マトリクスは、映像デコードと音声デコードの切り替え時に、必要に応じて QMEM 135 と DRAM 160 との間でスワップされる。

【0071】音声デコードのために、VLC/FLC デコード 120 は、6 つの量子化されスケール処理されたベクトル S を上記のように ZMEM 134 に書き込む。SPU 140 は逆量子化/逆スケール処理命令及びウィンドウ/マトリクス処理命令を ZMEM 134 の各ベクトル S 上で実行する。逆量子化/逆スケール処理命令は、ベクトル S に逆量子化、逆スケール処理、及びバタフライオペレーションを行うことによって、結合データ $T0 \sim T31$ を決定するものである。逆スケール処理のために、ZMEM 134 からの 16 ビット要素 S_i は MAC 750 の入力値 Z としてマルチプレクサ 716 を通して与えられる。マルチプレクサ 706 は 1 に等しい値 X を MAC 750 に接続されたレジスタ 707 に与え、マルチプレクサ 712 は式 6 で与えられた $K2$ に等しい値 Y を ROM 732 から MAC 750 に接続されたレジスタ 713 に与える。MAC 750 は値 X 及び値 Y の積から値 Z を引いた値を計算する。レジスタ 717 は、3 つの読み取りポート及び 3 つの書き込みポートを有するマルチポート式レジスタファイル 733 に書き込まれうる MAC 750 からの出力値をとらえる。値 $S_i + K2$ はレジスタファイル 733 に格納される。MAC 750 での 2 度目に通過に於いては、マルチプレクサ 706 は、レジスタファイル 733 からの $S_i + K2$ に等しい信号 X をレジスタ 707 を通して M

AC750に与える。マルチプレクサ712は、ROM732からのK1(式6)に等しい信号Yをレジスタ713を通してMAC750に与える。マルチプレクサ716は0に等しい値Zを与える。MAC750の出力信号は再びレジスタファイル733に書き込まれる逆量子化された値 S_i' である。

【0072】ROM732は交互にアクセスされるROM732A及び732Bを有し、ROM732全体でROM732A及び732B単体の2倍の読み取り速度を有することになる。ROM732は逆量子化、逆スケール処理、マトリクス処理、ウィンドウ処理及び映像デコードのための定数を有する。SPU140のコントロールユニットは、QMEM135のサイドインフォメーションからROM732に於ける正しいアドレスを定める。

【0073】逆量子化値 S_i' は、逆スケール処理のために値Xとしてマルチプレクサ706を通して与えられる。逆スケール処理は2つの乗算によって実施される。始めの乗算のために、マルチプレクサ712及びレジスタ713が、QMEM135のサイドインフォメーションからのインデックスに従って、ROM732から1、 $2^{-1/3}$ 、及び $2^{-2/3}$ の内の1つである第1スケールファクタF1を与える。マルチプレクサ716からの値Zは0である。部分的に逆スケール処理された値はレジスタ717によって保持され、レジスタファイル733に格納され、マルチプレクサ706及びレジスタ707を通して、値Xとして第2の乗算のために与えられることになる。マルチプレクサ712及びレジスタ713は、QMEM135のサイドインフォメーションからのインデックスに基づきROM732から 2^{-1} 〜 2^{-20} のうちの1つである第2のスケールファクタF2を与え、値Zは0である。F1とF2との積は式8のスケールファクタFに等しい。2つの乗算による逆スケール処理は小さいスケールファクタFによって行われる1回の乗算によって生ずるまるめ誤差を減らす。

【0074】逆量子化され、逆スケール処理された値 S_i はレジスタファイル733に書き込まれ、SPU140は、上記と同様にZMEM134からの第2要素 S_k の逆量子化及び逆スケール処理を行う。要素 S_k が逆量子化され逆スケール処理されて要素 S_k を与えるとき、バタフライユニット760は S_i 及び S_k の和と差を計算する。和と差の計算は他の要素の逆量子化及び逆スケール処理と平行して行われる。要素 S_i は参考資料Bに示されたT0〜T1の和と差の計算を簡単にするように逆スケール処理及び逆量子化される。

【0075】合計値T28を定めるための1つの例を示すと、要素S0、S31、S15、S16、S7、S24、S8、及びS23をこの順番に逆量子化及び逆スケールする。バタフライユニット760はS0及びS31の和と差を定める一方、MAC750が要素S15及び

S16を定める。レジスタ725は、レジスタファイル733に書き込むべく和 $S0+S31$ を保持する。レジスタ726は差 $(S0-S31)=T0$ を保持し、この値はレジスタ727、マルチプレクサ728、音声クランプ724、及びマルチプレクサ723を通過してTMEM136に書き込まれる。これに続いて、バタフライユニット760は差 $(S15-S16)=T15$ を計算し、この値はTMEM136に同様に格納され、また和 $(S15+S16)$ を計算しこの値はレジスタファイル733に一時的に格納される。次に、バタフライユニット760は和 $(S0+S31)$ 及び $(S15+S16)$ の和と差を計算する。差 $(S0+S31)-(S15+S16)=15$ はTMEM136にセーブされる。和 $(S0+S31)+(S15+S16)$ はレジスタファイル733に一時的に格納される。S1、S31、S15、S16についてなされたのと同じ計算がS17、S24、S8、及びS23についても行われて、 $(S7-S24)=P7$ 、 $(S8-S23)=P8$ 、 $(S7+S24)-(S8+S23)=P23$ 、及び $(S7+S24)+(S8+S23)$ が定められる。バタフライユニットは、次にレジスタファイル733からの値 $(S0+S31)+(S15+S16)$ 及び $(S7+S24)+(S8+S23)$ とを結合して、差 $T24[(S0+S31)+(S15+S16)]-[(S7+S24)+(S8+S23)]$ 及び和 $T28[(S0+S31)+(S15+S16)]-[(S7+S24)+(S8+S23)]$ を定め、両者はTMEM136に格納されることになる。ベクトルSの残りの要素は、参考資料Bの和T29〜T31を決定するのに必要な順番で、バタフライユニット760がそのオペレーションを行うのと平行して逆量子化される。

【0076】このように結合データT0〜T31が決定されてTMEM136に格納され、SPUはウィンドウ/マトリクス処理命令を実行する。結合データT0〜T31はマルチプレクサ706及びレジスタ707を通してMAC750に与えられる。MAC750は参考資料Cで与えられた、ROM732からのマトリクス処理係数と結合データT0〜T31との乗算を行い、要素V17〜V48を定める。バタフライユニット760は要素V17〜V48を蓄積するのに必要な加算または減算を行うが、要素V17〜V48はレジスタ726、727、マルチプレクサ728及びクランプ729を通過してPMEM137にセーブされることになる。必要な乗算の回数は、1つの要素Vi毎に、4、8、または16回でこれはインデックスの数によって決まる。

【0077】ウィンドウ処理によって、マトリクス処理によって得られたベクトル要素Viをフィルタ処理することになる。各ウィンドウ/マトリクス処理命令に対して、メモリコントローラ180は、DRAM160のメ

モリマップ 630にある場合 33 個の前ベクトル要素を読み取り、PMEM137に与えると共に、最も古いベクトル（ベクトル V^{-15} 及び V^{-14} ）からの 33 個の前ベクトル要素は、始めに DRAM160 から読み取られる。PMEM137 の前ベクトル要素はマルチプレクサ 706 及びレジスタ 707 を通して MAC750 へ与えられる。MAC750 は前ベクトル要素とウィンドウ処理係数 $D(i+32k)$ との積を計算し、この積を 32 の音声振幅値として TMEM136 に蓄積する。33 のベクトル要素の各セットに対して、64 の乗算が行われ、2 つの値は各音声振幅値に蓄積される。

【0078】ベクトルのマトリクス処理及びウィンドウ処理には 8 つのウィンドウ／マトリクス処理命令が必要である。各ウィンドウ／マトリクス処理命令はベクトル V^0 の 4 つの要素を決定して DRAM160 へ格納し、各 32 の音声振幅値 A_i に対する 2 つのウィンドウ処理の寄与を蓄積する。第 1 のウィンドウ／マトリクス処理命令の前に、古い音声振幅値 A_i は TMEM136 から引き出され、DRAM160 にセーブされなければならない。古い音声振幅値のセーブは、TMEM136 が二重ポート式である場合か、逆量子化を行う間に逆スケール処理を立ち往生 (stall) させて TMEM136 に書き込むことができる場合は、新しいベクトルの逆量子化及び逆スケール処理と同時に行うことができる。そうでない場合は、ウィンドウ／マトリクス処理命令は古い音声振幅値が DRAM160 にセーブされるまで待たなければならない。ウィンドウ／マトリクス処理命令は現ベクトルの逆量子化及び逆スケール処理が完了するまで待機しなければならない。

【0079】始めに、メモリコントローラ 180 は DRAM160 から 33 個のベクトル要素を PMEM137 の第 1 部分に転送する。MPEG 規格のためには、ベクトル要素は正確に 20 ビットに保持されなければならないが、標準的な DRAM は 16 ビットの記憶ロケーションを持っている。従って、33 個のベクトル要素は DRAM160 に於ける 42 個のアドレスに格納される。PMEM137 は、2 つの 9 ビット映像エラータームを保持するために 18 ビットの幅を有する。ベクトル要素は、DRAM137 から受け取られると、PMEM137 における 42 個のアドレスに格納される。33 個のベクトル要素が一旦 PMEM137 に格納されると、SPU は PMEM137 の第 1 部分に於いてウィンドウ／マトリクス処理命令の実行を開始し、必要に応じて 20 ビットの要素を検索する。ウィンドウ／マトリクス処理命令によって、PMEM137 における 33 個の要素のウィンドウ処理されたものが TMEM136 における音声振幅値 A_i に蓄積され、現ベクトル V^0 の 4 つのベクトル要素のセットを定める。各 4 つのベクトル要素のセット $V17 \sim V20$ 、 $V21 \sim V24$ 、 $V25 \sim V28$ 、 $V29 \sim V32$ 、 $V33 \sim V36$ 、 $V37 \sim V40$ 、 V

41 $\sim V44$ 、及び $V45 \sim V48$ が参考資料 C の式によって定められるならば 44 の乗算が必要であることに注意しなければならない。マトリクス処理によって定められる 4 つのベクトル要素のセットは PMEM137 に格納される。

【0080】ウィンドウ／マトリクス処理命令の実行と同時に、メモリコントローラ 180 は更に 33 個のベクトル要素を DRAM160 から PMEM の第 2 部分へ転送する。ウィンドウ／マトリクス処理命令が完了したとき、4 つのベクトル要素は PMEM137 から DRAM160 へ書き込まれ、そして次のウィンドウ／マトリクス処理命令が PMEM137 の第 2 部分を用いて開始される。ベクトル V^0 のための 8 番目の及び最後のウィンドウ／マトリクス処理命令はウィンドウ処理のためにベクトル V^0 の要素を用いる。ウィンドウ処理には要素 $V^017 \sim V^033$ だけが必要なので、ウィンドウ処理に必要な要素は、ウィンドウ処理のための検索を行う前に、前ウィンドウ／マトリクス処理命令に於いて計算され、DRAM160 に格納される。8 番目のウィンドウ／マトリクス処理命令の後、32 個の音声振幅値 A_i を DRAM160 に書き込むべく 16 ビットにクランプする。逆量子化及び逆スケール処理を立ち往生させられない場合は、SPU140 は、メモリコントローラ 180 が音声振幅値 A_i を DRAM160 に転送している間待機する。

【0081】乗算にかかる時間がマトリクス処理に限界を与え、ウィンドウ処理は DRAM160 へのメモリアクセスによって僅かに限界を与えられるので、これに関連して、一体に結合されたウィンドウ／マトリクス処理ステップには利点がある。ウィンドウ処理とマトリクス処理を結合することによって、SPU140 及びデコーダ 100 のリソースをより均等に使用する命令を与えることができる。更に、ウィンドウ処理とマトリクス処理が結合されていない場合は、DRAM160 から映像 FIFO171 及び 172 (及び FIFO125、115、190) への転送が、ウィンドウ処理を遅らせることになる。ウィンドウ処理とマトリクス処理とを結合することによって、マトリクス処理があまり多くの DRAM のバンド幅を取らないため、DRAM160 から映像 FIFO171、172 への転送が、ウィンドウ／マトリクス計算にオーバーラップすることができる。マトリクス処理には 4 つの 20 ビットの値を DRAM160 に書き込むことだけが必要なのである。

【0082】図 16 は、CPU150 によって音声データフレームのデコード 137 のために実行される処理ループを示したものであり、図 17 は、この処理ループのタイミングを示したものである。始めに、ステップ 805 に於いて、CPU150 は、層 1 音声データフレームまたは層 2 音声データフレームの一部のためにスケールファクタインデックス及びビットアロケーションを QM

EM135にロードする。次にステップ810に於いて、CPU150はメモリコントローラ180が33個のベクトル要素をDRAM160からPMEM137へ転送することを要求する。33個のベクトル要素は時間T1(図17)の間にPMEM137の第1の半分の領域へ転送される。その間に、CPU150は、ステップ820に於いて「サブバンド取り込み」コマンドを発し、時間T1の間に転送と平行してVLC/FLCデコーダ120がそれを実行する。「サブバンド取り込み」コマンドは上記のように、6つのベクトルの要素をZMEM134に移動させる。ステップ825に於いて、CPU150はVLC/FLCデコーダ120がアイドル状態となるまで待ち、その後「逆量子化/逆スケール処理」コマンドを、ステップ835に於いて発する。SPU140は、時間T1の間に転送と平行して「逆量子化/逆スケール処理」コマンドを実行する。

【0083】SPU140は、ステップ810に於いて必要な33個のベクトル要素の転送が終了する間に、「逆量子化/逆スケール処理」コマンドから「ウィンドウ/マトリクス処理」コマンドへ進むことができないが、これは33個のベクトル要素がウィンドウ処理に必要なからである。SPUは、マトリクス処理のために、「逆量子化/逆スケール処理」コマンド(ステップ835)の結果を必要とする。更に、「ウィンドウ/マトリクス処理」コマンドは、前に定められた音声振幅値がTMEM136から取り出されDRAM160へセーブされるまで実行を開始することができない。従って、CPU150は、ステップ840、845、及び855に於いて待機し、その後ステップ860に於いて、SPU140に、「ウィンドウ/マトリクス処理」コマンドを実行させる。時間T2の間、「ウィンドウ/マトリクス処理」コマンドは上記のように実行される。ステップ865に於いて、CPU150は、メモリコントローラ180に、更に33個のベクトル要素をDRAM160からPMEM137の第2の半分の領域へ転送することを要求する。時間T3の間に、ステップ86の「ウィンドウ/マトリクス処理」コマンドと平行して33個の更なる要素の転送が実行される。メモリコントローラ180はSPU140に用いられるデータに干渉しないが、これはメモリコントローラ180とSPU140とが、PMEM137のそれぞれ異なる半分の領域にアクセスするからである。

【0084】SPU140が「ウィンドウ/マトリクス処理」コマンドを終了するまで、CPU150はステップ870に於いて待機し、その後ステップ875に於いて、メモリコントローラ180に「ウィンドウ/マトリクス処理」コマンドによって定められた4つのベクトル要素をDRAM160へ転送することを要求する。DRAM160への転送は時間T4の間に行われる。CPU150は、ステップ880からステップ855に移行

し、ステップ865に於いて必要とされる33個の要素がPMEM137の第2の半分の領域へ転送されるまで待機する。デコードは上記のように進行するが、ステップ860の第2の「ウィンドウ/マトリクス処理」コマンドはPMEM137の第2の半分の領域で実行され、ステップ865の2度目の実行は、33個のベクトル要素をPMEM137の第1の半分の領域へ転送することを要求する点で異なる。SPU140は、第2の「ウィンドウ/マトリクス処理」コマンドを時間T5の間に実行し、これはT4の間に前の4つのベクトル要素をDRAM160へ転送するのと平行して行われ、かつその次に、図17に示すように、時間T6の間に次の33個のベクトル要素のセットをDRAM160から転送するのとも平行して行われる。

【0085】ステップ855～ステップ880は8回繰り返される。各反復に於いて、ステップ860及び865はPMEM137の第1及び第2の半分の領域上で交互に実施される。ステップ855～880の8回目の反復の間、ベクトルが最後のベクトルセットの中の最後のベクトルでなければ、ステップ865に於いて次のベクトルをウィンドウ処理するために、ベクトル要素がリクエストされる。「ウィンドウ/マトリクス処理」コマンドの8回の反復の後、CPU150はステップ885に移行して、32個の今定められた音声振幅値Aiを、TMEM136からDRAM136へ転送することを要求する。CPU150はステップ890へ進み、次にステップ835に進んでZMEM134の次のベクトルのために「逆量子化/逆スケール処理」コマンドの実行を開始する。ステップ835からステップ890のループは6回実行され、各2つのチャンネルの3つのベクトルをデコードする。6つのベクトルがデコードされた後、CPU150はステップ895からステップ820にジャンプして、次の6つのベクトルセットのためにサブバンドデータを取り込む。ステップ820からステップ895のループは、4つのベクトルセットのために実行される。6つのベクトルセットを4組処理した後、新しいビットアロケーション及びスケールファクタインデックスが必要となる。

【0086】参考資料DにはプログラムのCコードリストが含まれ、これは逆量子化、逆スケール処理、マトリクス処理、及びウィンドウ処理を上記のように実行するものである。

【0087】本発明の音声映像デコーダ100(図1)は、MPEG規格に基づき映像デコードも行う。MPEG規格の元での映像デコードは米国特許出願07/890,732号(平成5年特許願第151,595号)及び第07/669,818号明細書に記載されており、ここではそれらを参照されたい。VLC/FLCデコーダ125はデコーダFIFO125からの映像データストリームに於けるコードをZMEM134に格納された

量子化された離散コサイン変換(DCT)係数に変換する。映像デコードのために、ZMEM134はジグザグメモリと呼ばれることもあり、これは係数が格納される順番のためにこう呼ばれているのである。QMEM135は、音声デコードの後にDRAM160からQMEM135へスワップされ、または映像データストリームに従って変えられる逆量子化係数を保持する。SPU140は、DCT係数を逆量子化するために逆量子化定数を用いる。

【0088】SPU140は逆量子化されたDCT係数とコサインファクタとを乗算し、次にDCT係数を、二次元逆離散コサイン変換(IDCT)によって画素値に転換する。二次元IDCTは2つの一次元IDCTとして用いられてもよく、TMEM136はIDCTの間、計算途中の値を保持するのに用いられる。IDCTの後、生成されたエラータムはPMEM137に格納され、DRAM160に書き込まれる。デコードされた映像は、映像バス176に出力するために、ブロック171~175を通してDRAM160から読み取られる。

【0089】SPU140は逆量子化、コサイン乗算、及びIDCTを含むオペレーションを、米国特許出願第07/890,732号(平成5年特許願第151,595号)の明細書に記載されているように実行する。図12、図13に示されたブロックに加えて、SPU140は映像デコードの間、図14および図15に示された回路ブロックを用いる。コサイン乗算オペレーションを行う間、マルチプレクサ712は、MAC750とDCT係数とを積であるコサインファクタをROM732から選択するようにセットされる。逆量子化命令のために、逆量子化定数はマルチプレクサ714及びレジスタ715を経てQMEM135から検索される。マルチプレクサは、QMEM135から、16ビットの信号の最上位の8ビットまたは最下位の8ビットのどちらかを選択する。乗算器711はマルチプレクサ710によって与えられた値により逆量子化定数をスケール処理する。マルチプレクサ710は内部マクロブロックのDCTタムのための定数、若しくはレジスタ708及び709からの5ビットのスケール処理ファクタのどちらかを選択する。乗算器711によって、マルチプレクサ712及びレジスタ713を通してスケール処理された逆量子化定数がMAC750へ与えられ、ZMEM134からの検索されたDCT係数との乗算が行われる。

【0090】ZMEM134からの各9ビットのDCT係数は、MPEG規格の必要に従って、MAC750へ与えられる前にパディングされたり、デクリメンタ(decrementer)704によって減少させられたり、ラウンダ(rounder)733によって奇数または0に丸められたりしてもよく、若しくはクランプ705によって予め定められたレンジにクリッピングされても良い。ANDゲート702はZMEM134からの9ビットのDCT

係数を「コード化された」制御信号に応じて0にセットする。映像逆量子化命令の間、マルチプレクサ703は、右側のゲート702からの9ビットのzQCode[8:0]をパディングすることによって形成された11ビットの信号に等しい出力信号decrin[10:0]を選択する。この代わりに、逆量子化命令以外の命令を実行するとき、マルチプレクサ703は、レジスタファイル733からの11ビットの信号SRC3[13:3]に等しいdecrin[10:0]を選択する。デクリメンタ704は、MPEG規格によって必要とされるとき、信号decrin[10:0]を減少させて、出力信号decrout[10:0]を与える。減少オペレーションが必要でない場合、信号decrout[10:0]は信号decrin[10:0]に等しい。

【0091】ラウンダ733は、MPEG規格によって必要とされる場合、信号decrout[10:0]の出力データのビット0(LSB)及び4を置き換える。ラウンダ733はZMEM134からのDCT係数が0の場合、信号decrout[10:0]を0にするが、そうでない場合は、被逆量子化命令(例えばコサイン乗算命令)の実行中に信号SRC3[13:3]を0にする。レジスタファイル733からの信号SRC3のビット[21:14]は、信号decrout[10:0]のプリフィクスとなり、これによって19ビットの信号CLAMPIN[18:0]を生成し、クランプ705に与える。クランプ705は、非逆量子化命令の実行中に、信号CRAMPIN[18:0]をクランプ処理して、-2047と+2047との間の値を有する14ビットの信号CLAMPOUT[13:0]にする。これとは別に、逆量子化命令の間に、クランプ705が入力信号を変えずに通過させることもある。次に信号CLAMPOUT[13:0]は、右側で0にパディングされて、信号XをMAC750へ送るのようにマルチプレクサ706及びレジスタ707を通して22ビットの信号を形成する。

【0092】MAC750は実行される命令によって、2つの数XとYとの乗算を行ったり、(例えば逆量子化命令またはコサイン乗算命令の場合)、または $X \times Y - Z$ で表される値を計算したり(例えばIDCT乗算-減算命令の場合)することが出来る。DCT係数は、ZMEM134と、TMEM136とから取り出されて、レジスタファイル733に与えられる。更に、MAC750からの計算結果の値は、オペランドとしてレジスタファイル733をバイパスし、バタフライユニット760へ進ませることも出来る。

【0093】バタフライユニット760は、2つの入力オペランド及びYの和と差を同時に計算する。MAC750及びバタフライユニット760は、乗算命令の実行中にそれぞれのオペランド上で平行してオペレーション

を行うことができるので、乗算命令は、乗算の計算結果とバタフライユニットのオペレーションの結果を同時に生成することができる。更に、MAC750の出力値（「計算途中」の値）を、直接マルチプレクサ718を通してバタフライユニット760に与えるように用いることによって、パイプライン処理を行うことができる。このような形にすることにより、スループットを高めることができるが、これはレジスタファイル733に於ける計算途中の値をロードし、次に読み込むことで起こる遅れがなくなるからである。

【0094】第1パスIDCTのバタフライオペレーションの結果はTMEM136に流されるが、これに対して第2パスIDCTオペレーションのバタフライオペレーションの結果はクランプ729によって「クリッピング」され、PMEM137へ流される。映像デコードのためにSPU140によって実行されるプログラムは、米国特許出願第07/890、732号に記載されている。

【0095】MPEG規格は音声ビットストリームに注入されるエラーコードを画定しないが、これはどのようなビット結合データも、音声データフレームの層1または層2のビットストリームに於いて有効に現われ得るからである。この代わりに、検出されたエラーを含む如何なる音声データバイトについても、CD-DSPは分割されたエラー信号を生成する。図1の音声映像デコード100は、音声/エラーコードインジェクタ118を有するが、これはエラー信号をCD-DSPから受け取ったとき、シリアルバス104上で受け取られた音声データの部分を、音声データフレームに於いてまれにしかみられないビット結合データに変化させるものである。コードFIFO115はエラー信号に対し9番目のビットと共に、2バイトのコード化されたデータを格納するべく18ビットの幅を持つ。音声エラーコードインジェクタ118はコード化されたデータのエラービットを探し、エラービットがセットされている場合は、エラーコードでそのバイトを上書きする。例えば、音声データストリームを受け取っている間にデコード100がエラー信号を受け取った場合は、音声エラーコードインジェクタ118は、音声データフレームに、16進法で7FFD7FFDの整合された値の32ビットワードを挿入する。この場合、エラーコードはエラー及び3つの他のバイトでこのバイトを置き換える。実際の、音声のためのMPEGデータストリームの統計学研究によれば、この32ビットの値\$7FFFD7FFDは、音声データの中で100時間に1回の割合よりも小さい発生率しか有さない。

【0096】エラーを有するバイトは、コードFIFO115に書き込まれた時はエラーコードで上書きされることができないが、これはデータストリームの型が異なれば、異なったエラーコードを用いているためであり、

デコード100はデータがコードFIFO115から取り除かれるまでエラーを含んだデータストリームの型を同定することができないからである。例えば、音声映像デコード100は、音声データストリーム、映像データストリーム、及びリリック (lyric) データストリームを受け取ってもよい。映像データストリームに於けるエラーは、\$000001B4で上書きされる。リリックデータストリームに於けるエラーは、32ビットの0で上書きされる。

10 【0097】音声データフレームのためのエラー信号を受け取ったときは、ホストインターフェイス110が8ビットシフトレジスタに1を挿入するが、このシフトレジスタは全ての音声データパケットに対して1度だけシフトされるものである。従って、シフトレジスタに於ける値は多くの音声データフレームに対して0ではなく、シフトレジスタから1を移動させるのに必要なシフト数と等しいかそれより大きい数である。シフトレジスタに於ける値は、DRAM160に於ける入力音声データバッファがエラーコードを含むことができる時間に対して0でない値をとる。VLC/FLCデコード120は、デコードされた全ての音声フレームに於けるエラーコードに等しいビット結合データを捜すが、このときシフトレジスタに於ける値は0でない。ビット結合データが検出された場合は、VLC/FLCデコード120はエラー隠蔽プロシジャ (error concealment procedure) を初期化する。実際に注入されたエラーコードでないビット結合データはめったに検出されないが、これはビット結合データが短時間の間にエラーを含む音声フレームを生起させる確率が小さいからである。

30 【0098】エラー隠蔽プロシジャは、データストリームに於けるエラーが音声の質に与える影響を最小化しようとする。例えば、サブバンドデータに於いてエラーコードが発生した場合は、VLC/FLCデコード120がエラーコードで損なわれた要素を0で置き換え、これにより発生した音声はごく1部の周波数成分を失うだけですむ。エラーコードによって音声データフレームのサイドインフォメーションまたはヘッダが損なわれ、音声データフレームがデコードされ得なくなってしまう場合は、VLC/FLCデコード120は割り込み信号を発生してCPU150へ送る。CPU150は、前音声データフレームを用いて失われたデータを再構成するか、SPU140にそのチャンネルの前音声データフレームを再びデコードさせるようにすることができる。

【0099】本発明の特定の実施例について述べてきたが、これは本発明の応用例の1つに過ぎず、これに限られるものではない。ここに開示した実施例の様々な改変を本発明の範囲を逸脱することなく行い得ることは当業者には明らかであろう。

【0100】

50 【発明の効果】従って、本発明により、回路の量を減ら

した効率的な音声映像デコーダ及び音声映像デコードの方法が提供される。

【図面の簡単な説明】

【図1】本発明の実施例の1つに基づくMPEG音声映像デコーダのブロック図を示したものである。

【図2】本発明の実施例の1つに基づく逆グループ処理回路のブロック図を示したものである。

【図3】本発明に基づく3を除数とする除算回路のブロック図を示したものである。

【図4】本発明に基づく3を除数とする除算回路の真理値表を示したものである。

【図5】本発明に基づく3を除数とする除算回路のゲートレベル図を示したものである。

【図6】本発明に基づく5を除数とする除算回路のブロック図を示したものである。

【図7】本発明に基づく5を除数とする除算回路の真理値表を示したものである。

【図8】本発明に基づく5を除数とする除算回路のゲートレベル図を示したものである。

【図9】本発明に基づく他の実施例の、逆グループ処理回路のブロック図を示したものである。

【図10】本発明に基づく他の実施例の、3、5、若しくは9で除す除算を行うためのアドレスジェネレータの、ゲートレベル図を示したものである。

【図11】本発明に基づき、ウィンドウ処理の間に用いられる前ベクトル要素のメモリマップを示したものである。

【図12】本発明の実施例の1つに基づく信号処理ユニットの実施例の1つのブロック図を示したものである。

【図13】本発明の実施例の1つに基づく信号処理ユニットの実施例の1つのブロック図を示したものである。

【図14】本発明の実施例の1つに基づく信号処理ユニットの実施例の1つのブロック図を示したものである。

【図15】本発明の実施例の1つに基づく信号処理ユニットの実施例の1つのブロック図を示したものである。

【図16】本発明に基づく音声信号デコード処理の流れ図を示したものである。

【図17】図16の処理のためのタイミング図を示したものである。

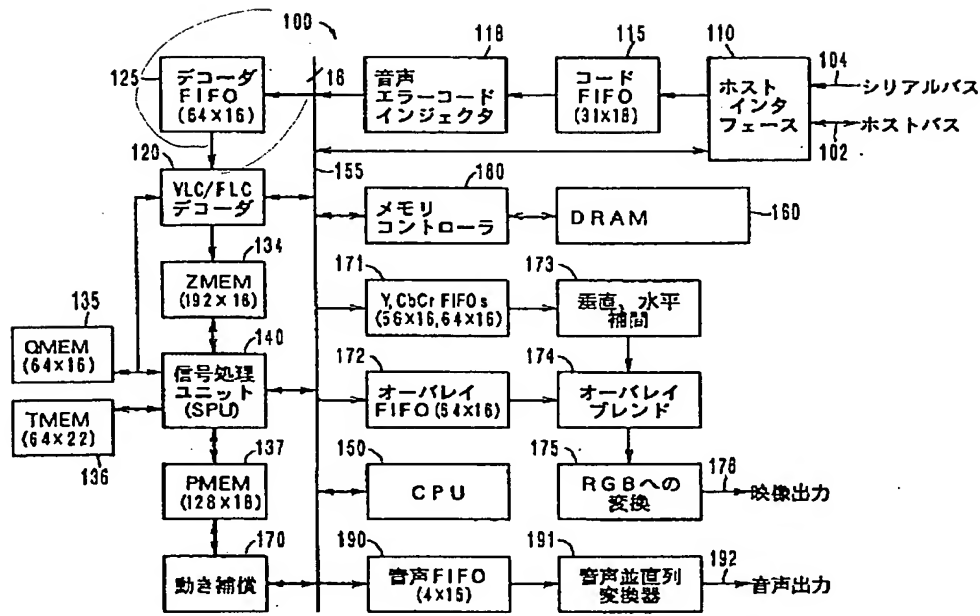
【符号の説明】

- 100 音声映像デコーダ
- 102 ホストバス
- 104 シリアルバス
- 115 コードFIFO
- 118 音声エラーコードインジェクタ
- 120 VLC/FLCデコーダ
- 125 デコーダFIFO
- 134～137 スタティックランダムアクセスメモリ (SRAM)
- 140 信号処理ユニット (SPU)

- 150 CPU
- 155 メインバス
- 160 ダイナミックランダムアクセスメモリ (DRAM)
- 170 動き補償回路ブロック
- 171 映像FIFOブロック
- 172 オーバレイFIFOブロック
- 173 補間器
- 174 ブレンド処理ブロック
- 175 コンバータ
- 176 映像バス
- 180 メモリコントローラ
- 190 出力音声FIFO
- 191 音声並直列変換器
- 192 音声バス
- 200 デコード回路
- 210 除算器
- 220～221 マルチプレクサ
- 230～232 レジスタ
- 300 除算回路
- 310、320 回路ブロック
- 311、321 被除数バス
- 312 余りバス
- 400 除算回路
- 410、420、430 回路ブロック
- 411、421、431 被除数バス
- 412 余りバス
- 413、423、433 商出力バス
- 434 余り出力バス
- 500 デコード回路
- 550 リードオンリメモリ (ROM)
- 560 アドレスジェネレータ
- 610、620、630 メモリマップ
- 702 ANDゲート
- 703 マルチプレクサ
- 704 デクリメンタ
- 705 クランプ
- 706、712、714、716、718、723 マルチプレクサ
- 707、713、715、717、725、726、727 レジスタ
- 711 乗算器
- 724 音声クランプ
- 728、730 マルチプレクサ
- 729 映像クランプ
- 732 ROM
- 733 レジスタファイル (ラウンダ)
- 750 MAC
- 760 バタフライユニット

【図 1】

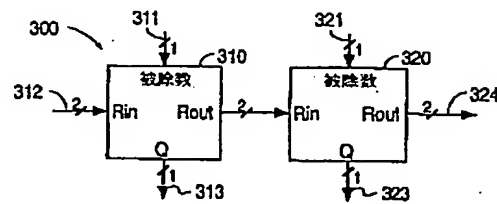
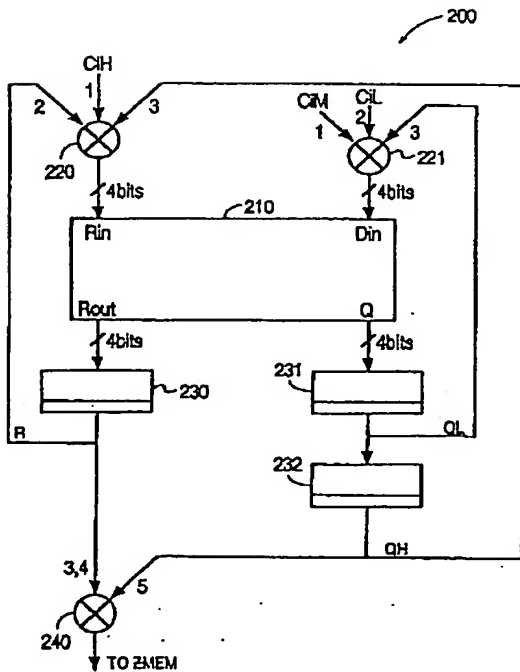
【図 7】



Rin				被除数	商	余り
A	B	C	Din	Q	Rout	T
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	0	1	0
0	1	0	1	1	0	0
0	1	1	0	1	0	0
0	1	1	1	1	0	1
1	0	0	0	1	0	1
1	0	0	1	1	1	0

【図 2】

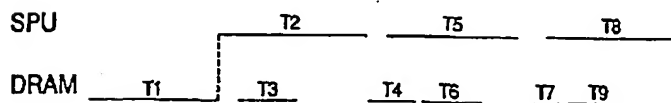
【図 3】



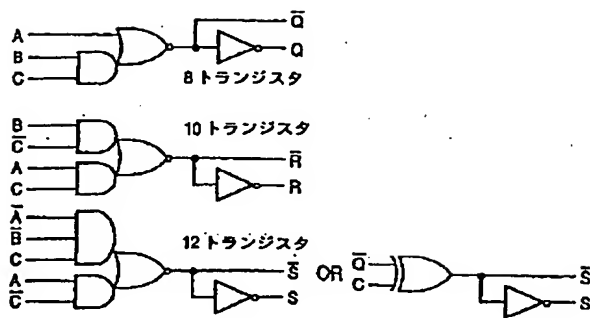
【図 4】

Rin			被除数		Rout		
A	B	C	Q		R	S	
0	0	0	0		0	0	
0	0	1	0		0	1	\overline{ABC}
0	1	0	0		1	0	BC
0	1	1	1		0	0	$\left. \begin{array}{l} \overline{AC} \\ AC \end{array} \right\}$
1	0	0	1		0	1	
1	0	1	1		1	0	
1	1	0	x		x	x	
1	1	1	x		x	x	

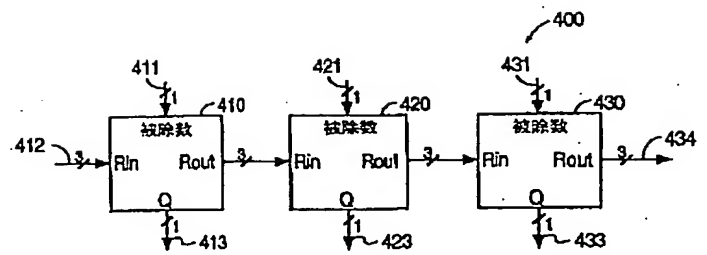
【図 17】



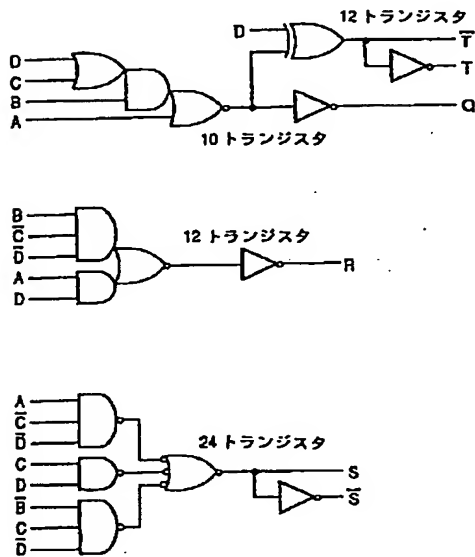
【図 5】



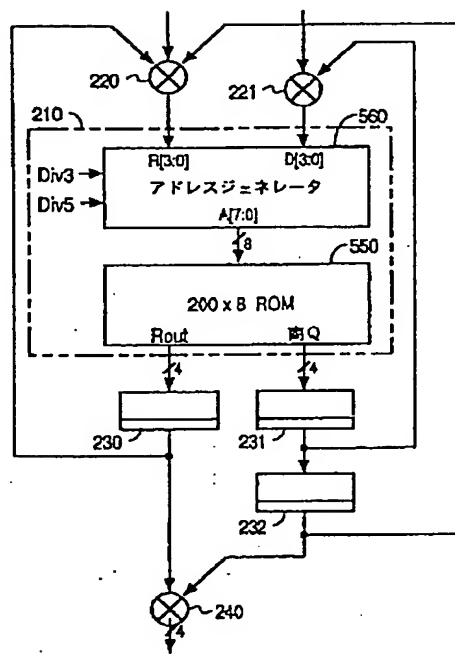
【図 6】



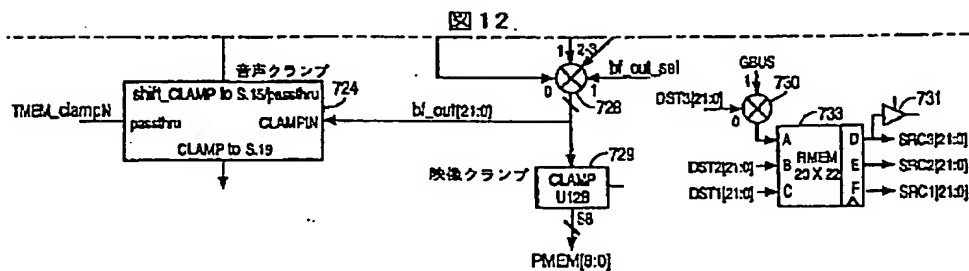
【図 8】



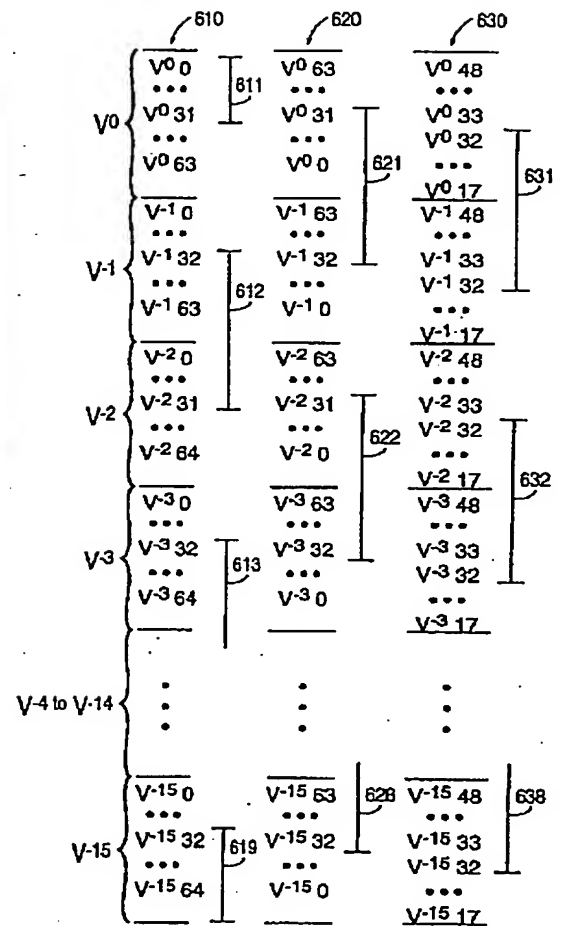
【図 9】



【図 13】



【图 1 1】



【図 15】

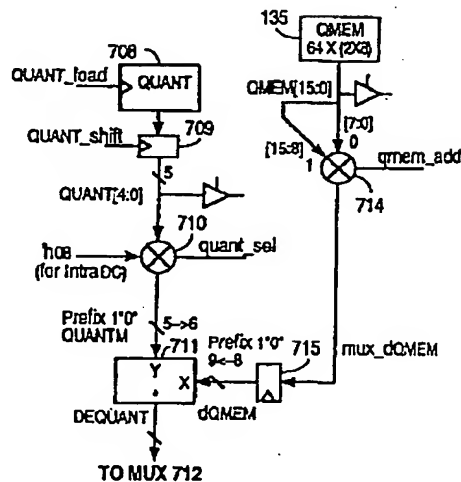


図 14 よう

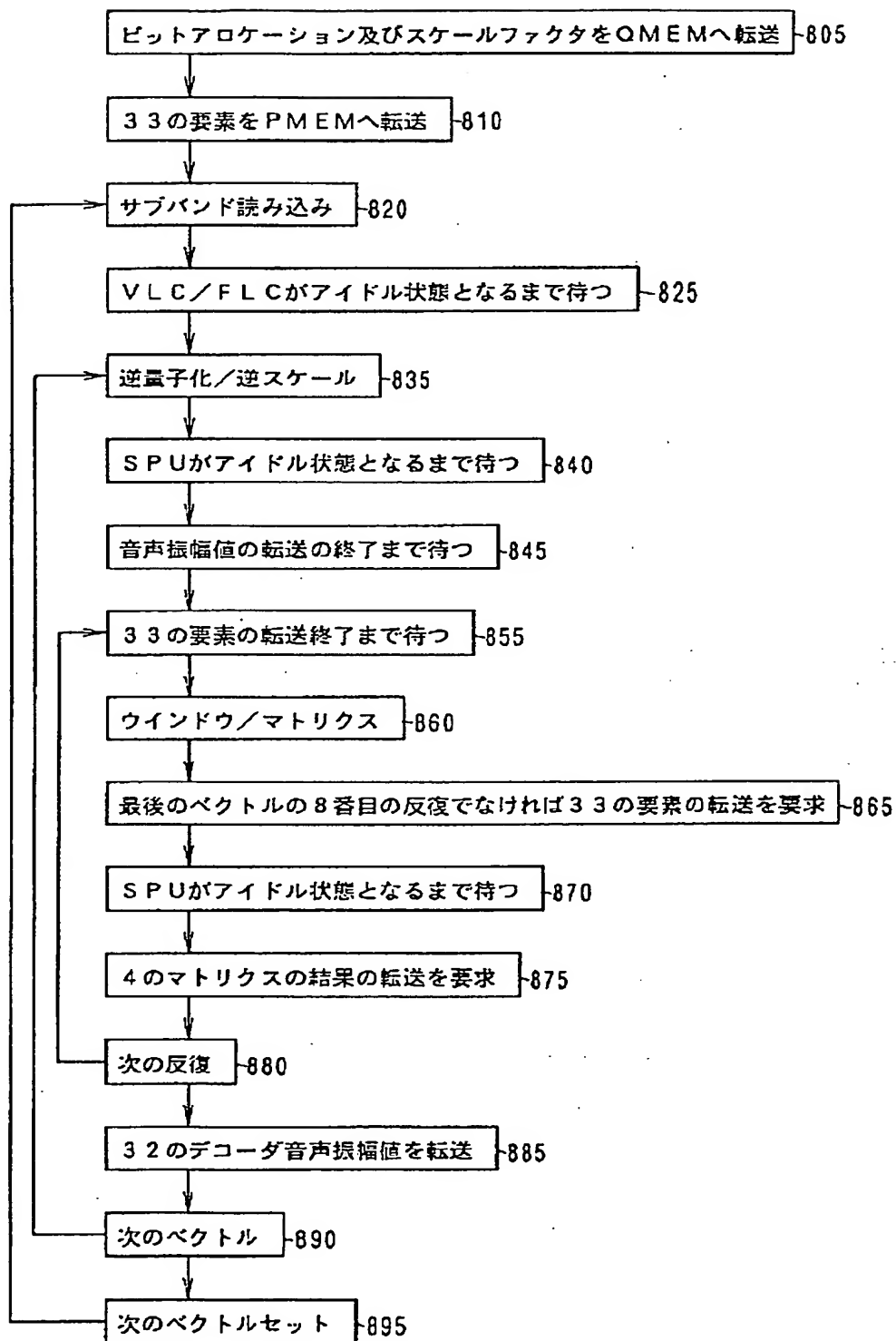
140

図 15 よう

Figure 14 is a block diagram of a video processing system. The diagram shows a sequence of operations starting from input signals (SRC3, dTMEM, dpMEM) through various multipliers (706, 713, 716, 717, 718, 722, 723) and adders (707, 720, 725, 726, 727). The system includes memory blocks (TMEM, dTMEM, CROM) and a final output stage (723, 725, 726, 727) producing DST1 and DST2 signals.

Figure 15 is a block diagram of a video processing system. The diagram shows a sequence of operations starting from input signals (SRC3, dTMEM, dpMEM) through various multipliers (706, 713, 716, 717, 718, 722, 723) and adders (707, 720, 725, 726, 727). The system includes memory blocks (TMEM, dTMEM, CROM) and a final output stage (723, 725, 726, 727) producing DST1 and DST2 signals.

【図 1 6】



フロントページの続き

(51)Int. Cl.⁶

識別記号

庁内整理番号

F I

技術表示箇所

H 0 4 N 5/92

7/24

H 0 4 N 7/13

Z